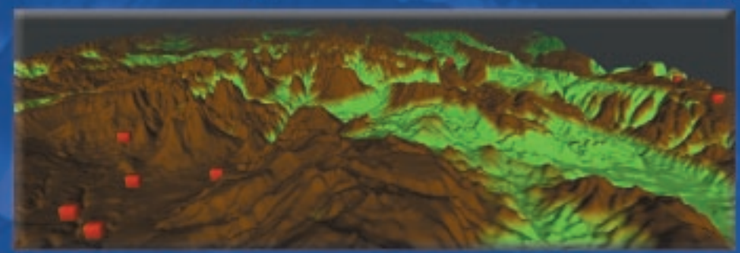
Large-Scale Collective Communication and Load-Balancing on Parallel HPC Systems

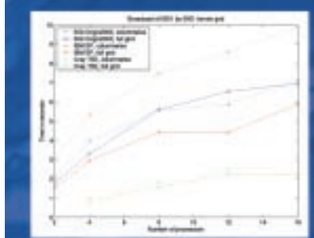


A terrain map is displayed with threats. Green shows the zones immediately above terrain that it is safe for aircraft to remain undetected by threats. Yellow also represents safe areas, but where the ceiling under which aircraft must fly is close to the ground. The red cubes represent threats, e.g., radar or antiaircraft batteries.

Mark R. Fahey

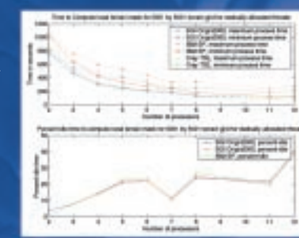
U.S. Army Engineer Research and Development Center

Major Shared Resource Center



This plot shows the time in seconds to perform a broadcast of a 6001 by 6001 terrain grid (integer valued) to all processes.

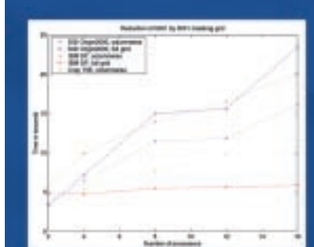
Terrain masking is the process of determining whether a location is masked from threats by intervening terrain. Threats could be light sources, radio towers, antennae, radar receivers, etc. Terrain masking computes the safe altitude, called the masking altitude, at each grid point. Aircraft flying above this altitude can be observed by at least one enemy sensor. At each grid point within range of a threat, the masking height associated with that threat is computed. The final masking height at each grid point is the minimum of the computed masking altitudes associated with each threat covering the grid point.



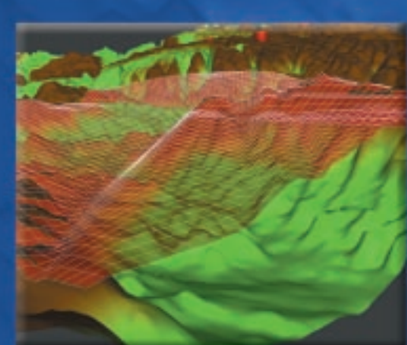
(Top) The time in seconds to compute local terrain mask for statically allocated threats. The maximum and minimum timings of MPI processes are shown. (Bottom) The percent idle time is shown. Percent idle time = 100% - $t_i / (N * t_{max})$ where N is the number of MPI processes and t_i is the execution time for process i.

The masking heights corresponding to each threat can be computed concurrently. For distributed memory computers, each process must have a local copy of the masking grid, and a global reduction must be performed after all threats have been processed. Terrain masks can be computed by the following steps on a distributed memory parallel computer: master MPI process reads terrain grid; terrain grid is broadcast to worker processes; the threats are partitioned among the processes; each process computes the local terrain mask for its assigned threats; and a global reduction of masking grid is performed on master process.

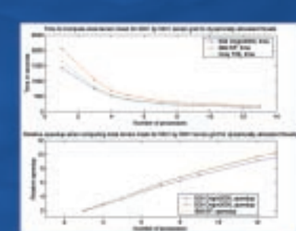
The time spent computing the terrain mask for each threat can vary greatly. Statically allocating the threats to processes results in poor load-balancing. This shortcoming can be remedied by dynamically assigning work to processors. Of course this strategy has the drawback that one process must monitor the worker processes and does no computational work.



A plot showing the time in seconds to perform a reduction of a 6001 by 6001 masking grid (integer valued) over all processes.



A portion of the above terrain map is enlarged with part of the mask shown. The white canopy represents the maximum height under which flying objects cannot be detected by the threats located on the terrain map.



(Top) The time in seconds to compute local terrain mask for dynamically allocated threats. (Bottom) The relative speedup is shown. Relative speedup = T_1/T_p where T_1 is the execution time on 1 processor and T_p is the execution time on p processors.